Data Security Controls and Procedures— A Philosophy for DP Installations

IBM

G320-5649-01

Data Security Controls and Procedures – A Philosophy for DP Installations

This publication is addressed to information systems management and presents measures that IBM and others have found useful for limiting risk in data processing installations. While it is hoped that these measures will be of interest, they are provided for information purposes only, and IBM makes no guarantee or representation with respect to their applicability to a specific data processing installation.

Second Edition (March 1977)

This edition is a major revision and obsoletes the previous edition.

Requests for copies of IBM publications should be made to your IBM representative or to the IBM branch office serving you locality.

A form for readers' comments has been included at the back of this publication. If this form has been removed, address comments concerning the contents of this publication to IBM Corporation, Technical Publications/Systems, Dept. 824, 1133 Westchester Avenue, White Plains, New York 10604. Comments become the property of IBM.

Preface

This publication is addressed to information systems management, and it presents measures that IBM and others have found useful for limiting risk in data processing activities. While it is hoped that these measures will be of interest, they are provided for information purposes only, and IBM makes no guarantee or representation with respect to their applicability to a specific data processing installation. It attempts to answer questions such as:

- How are security requirements reconciled with production needs?
- How can control be exercised over system programming activities? operations? system development?
- Who should be made responsible for security?

Emphasis is on controls and procedures as they relate to data security; their use in the accomplishment of other objectives is treated only in passing. Physical security measures and backup and recovery measures are dealt with only as they relate to planning. While some variation of most of the techniques discussed should be applicable to most organizations large enough to have a data processing system, they should be carefully selected and applied so that they will be both cost-effective and consistent with all of the organization's objectives and policies.

Chapter 1 discusses the objective commonly called "security" and shows how controls and procedures contribute to it. Chapters 2 and 3 treat the properties and functions that an information system must have if it is to be operated with an acceptable level of risk. These chapters should be useful to system designers as well as management.

Chapter 4 covers the traditional classes of control such as organizational, operational and personnel controls as well as development and documentation controls useful with data processing systems. While the primary audience of this chapter is management, auditors and system designers may also find it useful.

Chapter 5 presents a model for analyzing a job or position in order to determine appropriate controls. Several typical positions in data processing are then examined in terms of the model and suitable controls are suggested.

Plans and programs that high level general management may find useful for reducing risk are presented in Chapter 6, with common techniques for implementing them.

While it is recognized that this structure involves some redundancy, it was adopted to provide a greater degree of completeness and clarity.

Contents

Chapter 1. Introduction . . . 1

Chapter 2. System Properties . . . 2
Integrity . . . 2
Test of Predictability . . . 2
Auditability . . . 2
Sufficient Conditions for Auditable Systems . . . 2
Tests of Accountability and Visibility . . . 2
Controllability . . . 3
Tests of Granularity and Specificity . . . 3

Chapter 3. Interface Controls . . . 4
Identification . . . 4
Authentication . . . 4
Passwords . . . 4
Keys and Magnetic-Stripe Cards . . . 4
Authorization . . . 5
Delegation . . . 5
Journaling . . . 5
Variance Detection . . . 5
Summary . . . 6

Chapter 4. Traditional Classes of Control . . . 7
Organizational Controls . . . 7
Separate DP from Users . . . 7
Separate Duties within DP . . . 7
Maintain Traditional Separations . . . 7
Assign Individual Security Responsibility . . . 8
Use Appropriate Span of Management Control . . . 8
Use Security Staff Where Appropriate . . . 8
Personnel Procedures . . . 8
Hiring Procedures . . . 8
Control Vacations and Job Rotation . . . 8
Restrict Employee Accounts . . . 9
Follow Complete Termination Procedures . . . 9
Operational Controls . . . 9
Compare Output with Input . . . 9
Use Conversational Feedback . . . 9
Use Redundant Data . . . 10
Use Control Totals . . . 10
Control Error Corrections . . . 10
Supervise Recovery and Restart . . . 10
Control Forms . . . 10
Control Job Input/Output Media . . . 11
Development Controls . . . 11
View Programming as an Application . . . 11
View Program Library as a File . . . 12
Involve Multiple People in Changes to Library . . . 12
Control Access to the Program Library . . . 12
Log All Changes to Program Library . . . 12
Structure Library and Fix Responsibility . . . 12
Use Control Data on Program Library . . . 12
Specify Content of Library to Facilitate Reconciliation . . . 12
Compare Additions to Library with Authorizations . . . 12
Compare Programs with Expectations . . . 13
Compare Program Changes with Authorizations . . . 13
Compare Library and Programs with EarlierVersion . . . 13

Chapter 5. Functional Duties . . . 14
Computer Operator . . . 14
Application Development Programmer . . . 14
Maintenance Programmer . . . 15
Systems Programmer . . . 15
Magnetic Storage Media Librarian . . . 16
Conclusion . . . 16

Chapter 6. Plans and Programs 17
Risk Assessment 17
Emergency Plan 17
Backup Plan 17
Identify Critical Jobs 17
Identify Critical Configuration 17
Identify Backup System 17
Assign Backup Missions 18
Assign Individual Roles 18
Provide for Security in Backup Situation 18
Test Backup Plan 18
Recovery Plan 18
Make Assignments and Plans for Communicating Them 18
Identify Alternate Sites 18
Identify Source of Replacement Equipment 19
Plan for Replacement Communication 19
Facilities 19
Provide for Availability of Data, Supplies, and Recovery Plan 19
Test Recovery Plan 19
Vital Records Plan 19
Access Control Plan 20
Localize Control 20
Use Multiple Control Points 20
Use Appropriate Technology 20
Note Variances and Follow Up 20
Data Classification Program 20
Establish Classification Rules 21
Restrict Classification Authority 21
Data Handling Policies 21
Keep Sensitive Data under Lock and Key 21
Provide for Secure Waste Disposal 21
Provide for Secure Distribution and Safekeeping 21
Control Copying and Reproduction 21
Education 22
Auditing Program 22

Chapter 7. Conclusion 23

Bibliography 24

Chapter 1. Introduction

Security may be defined as the condition of safety of people, facilities, and data from natural and man-made hazards. These hazards include fire, wind, rain, rising water, lightning, and earthquake; error, mischief, vandalism, riot, war, conversion, misappropriation, fraud, embezzlement, or theft.

Data security is the subset of that problem dealing with the single resource – data. While the same hazards apply to data, there can be only six adverse effects: modification, destruction, or disclosure, either accidental or intentional.

Evidence suggests that accidental events account for the most damage because they are frequent, although the consequences of the *average* accidental event are limited. Intentional acts tend to have more serious consequences per event, but they occur less often.

Protective measures to reduce these exposures may be placed in three categories:

Physical measures
Backup and recovery measures
Controls and procedures

Physical measures include: (1) providing a safe place to work; (2) controlling access to sensitive resources such as cash and securities or the computer; (3) detecting and communicating emergencies such as fires or intrusions; and (4) limiting the damage resulting from emergencies. Among these measures are safe buildings, doors, locks, file cabinets, lock boxes, tape libraries, closed circuit TV, vaults, cash registers, fire alarms, burglar alarms, fire extinguishers, quenching and flooding systems, protective coverings, sump pumps and smoke removal systems.

Backup and recovery measures limit the consequences of an event that has already taken place. Backup includes measures designed to ensure that time-dependent portions of the mission can be achieved even though the primary mission capability has been lost. Recovery includes the measures to restore the primary mission capability.

Controls and procedures, the subject of this document, contribute to security by helping to ensure that planned events do happen and unintended events do not happen. "Control" has been defined as a mechanism used to regulate or guide the operation of a system; and as the direction, regulation, and coordination of a business activity. "Procedure" has been defined as a specific way of doing something, and as a series of steps followed in a regular, definite order. "Controls and procedures" as used here is intended to mean that collection of management tools and routines intended to direct and guide the activity in the direction of management objectives in general and security objectives in particular.

The computer contributes to security by helping to achieve timely, uniform and consistent application of controls and procedures. This subject will be viewed from five perspectives:

System Properties
Interface Controls
Traditional Classes of Control
Functional Duties
Plans and Programs

Chapter 2. System Properties

The first step in ensuring that an information system can be operated with an acceptable level of risk is to design it with that objective in mind. Such a system should manifest the following properties:

Integrity
Auditability
Controllability

Integrity

In the broadest sense, integrity may be defined as "that which is essential to wholeness or completeness." A system may be said to manifest the property of integrity if it performs according to its specification. For complex systems, the specification must include statements about how the system will fail. For example, an airliner's specification must state not only the attitudes in which it will fly, but also how it should behave if it is in an unusual attitude, so that the pilot knows what corrective action to take. For a computer system, the specification must state how the system will perform when all of its components are functioning properly, and also what malfunctions it can detect and how it will behave in the presence of malfunctions. The integrity of a system may be provided by the integrity of its components (this would apply to a jet engine). In theory, this kind of integrity can be demonstrated logically, by inspection or by testing. The more complex the system, however, the more difficult it is to achieve component integrity.

For this reason, complex systems are designed to have functional integrity. In other words, they are designed in such a way that the failure of a single component will not cause the whole system to fail. An airliner has this kind of integrity: the failure of the automatic pilot can be compensated for by use of the manual controls; the failure of a single tire or engine, or even both at once, should not result in the failure of the system.

An information system, like any system, manifests the property of integrity by performing – which includes failing – according to its specification. The specification should ensure that the system will fail in a limited and non-destructive manner, attempt corrective action, and, if corrective action cannot be taken, provide a positive indication of the failure, its type and its location.

Test of Predictability

Predictability is a useful test for integrity. The system may be said to manifest the property of integrity if for every stimulus to the system the response can be predicted both when the system is performing properly as well as when it is failing.

For example, a program has integrity if the response to all anticipated inputs is the expected output. For unanticipated inputs, the result must still be predictable (e.g., error message or return code). If the program does not always give the same output for a given input, or if its output for unanticipated inputs is unpredictable, then it lacks integrity.

Auditability

Auditability may be defined as the property that provides for relative ease in examining, verifying, or demonstrating the system. It must be possible for this determination to be made by individuals who are independent of the system. In the specific case of an information system, it must be possible to determine that the system is being used as intended, that the application system itself conforms to standards of good practice, and that the data contained in the application system conforms to expectation (e.g., accurate, complete, in conformity with the environment).

Sufficient Conditions for Auditable Systems

Auditability may be achieved in various ways for a simple system, but a complex system must meet the following conditions:

1. It must be composed of auditable subsystems, to a primitive or elemental level (e.g., a computer is composed of a central processing unit (CPU), channels, independent or integrated control units, and input/output units).
2. The subsystems must communicate with each other only across interfaces of specified format and content (e.g., a terminal communicates with storage *only* via communication lines, control unit, channel, CPU, channel, control unit, storage device).
3. There must exist the ability to record all of the transactions (events and content, stimulus and response) at the interfaces.
4. These records must include adequate reference to the external environment (e.g., time, place, user, authority).

Tests of Accountability and Visibility

For a system to be auditable, it must meet the tests of "accountability" and "visibility". The test of accountability is that it must be possible to fix responsibility for every significant event to the level of a single individual. The test of visibility is that a variance from the expected use of the system must come to the attention of responsible management in such a way as to permit timely and appropriate corrective action. It should be noted that these two tests limit each other. The test of accountability suggests recording a great deal of data, but recording too much data may obscure visibility.

For a 50,000-instruction computer program to be auditable, it would be helpful for it to divide into subprograms, composed of modules, composed of lines of high-level source code, which may generate machine language instructions implemented in the machine. Communications between these subprograms and modules should take place only in clearly specified ways; at a minimum, the communications that involve passing information from one person's control to another must be recorded.

Each program must conform to a functional specification that relates it to the environment (files, users, other programs, etc.). The labels, references and comments used in the program should be clear and meaningful. Listings should be readable and meaningful to others.

Controllability

Controllability is the property of a system that permits management to exercise a directing or restraining influence over its behavior, use, or content. For complex systems it is necessary for each system and sub-system to be individually controllable. For example, in an information system each subsystem must be able to control its own domain in such a way that it passes to other systems only those resources that it intends. More specifically, a computer system in a bank should be able to control its domain (and the bank be able to control the computer) in such a way that one teller can process deposits and withdrawals, but not open or close accounts or change names and addresses. It might also be necessary to restrict the teller to a specific subset of accounts such as those associated with a particular branch.

Tests of Granularity and Specificity

This property must meet the tests of "granularity" and "specificity" if it is to enable the system user to limit risk to an acceptable level. The test of granularity requires that the size of the resource to be controlled be small enough to constitute an acceptable risk. For example, if a transaction included both the ability to add vendors to the payables file and to approve invoices for payment (normally separated duties), it would fail the test of granularity. The test of specificity requires that it be possible to predict the effect of passing a resource or combination of resources from one process to another. In the example above, if the name of the transaction were ADDNEWVENDOR, it might fail the test of specificity since it includes an effect that cannot readily be predicted from the name. Preferably, the prediction should be feasible with no more data than the name or label of the resource. Granularity contributes to specificity as does the assignment of descriptive and meaningful names to resources.

Chapter 3. Interface Controls

One strategy for achieving the properties of integrity, auditability and controllability is to place functional controls at each interface where a significant resource passes from one person's control to another's. These functional controls may include:

Identification
Authentication
Authorization
Delegation
Journaling
Variance Detection

Identification

The identification function provides the ability to have names for users and resources. It implies a record that associates an identifier, name, or label with the user or resource. The label should be as concise as possible while still being large enough to uniquely name all members of the set. It should be mnemonic and stable over the life of the association with the interface and should therefore be assumed to be public. In small user populations, names or initials may be used. In larger populations, employee serial numbers may be more appropriate. It will also be useful to know the names of sets (such as department number) of which the user is a member. However, department number should not be used in lieu of a more specific label because of the need for individual accountability.

Authentication

The authentication function provides the ability to verify the identifying data provided by a user at an interface. At a person-to-person interface, this verification is often by physical appearance. The appearance may be compared either with memory or a photograph. Where a record is required, the evidence is usually a signature. At a person/machine interface, this verification normally relies on a combination of the following:

- Something that only one user knows, such as a keyword or password
- Something that only one person has, such as a key or card
- Some unique physical characteristic such as a fingerprint or hand geometry
- Some unique but reproducible behavior, such as the way a person speaks.

Passwords

The password is the most widely used authenticator because it is the most economical. Passwords should be short and changed frequently so as to minimize the exposure associated with a compromise. The frequency of change is a function of the consequences of a compromise and its probability. The probability of compromise is a function of the care exercised by the user and the frequency with which the password is used.

In time sharing systems, passwords are usually changed by the user. This is appropriate since the user knows more about the sensitivity of the data associated with the password and about its frequency of use.

In data base systems, the password is usually changed centrally for all users at the same time. This assures the custodians of the system, who represent the owners of the data, that the passwords are being changed. However, it requires that there be a distribution of the new passwords, involving some risk of compromise that must be balanced against the risk of not changing. Changing the password monthly appears to be a reasonable solution for user populations numbering in the thousands.

The distribution procedures must provide adequate protection against compromise and provide some evidence that the right person received the password. Where the use of the password is fairly regular and the change frequency is at least monthly, a good technique is to always distribute the password on the same day. Users can then be instructed to notify management if the password does not arrive (evidence that it may have been intercepted) or if it arrives but appears to have been opened.

For systems where the use is less regular and the change frequency less than monthly, the new code may be accompanied by a turnaround document requiring a positive acknowledgement of receipt. These receipts must be reconciled and variances pursued. Since time is required for this, this technique does involve some exposure.

Keys and Magnetic-Stripe Cards

For large user population and sensitive resources, it is desirable to use a key or magnetically encoded card in addition to the password. Keys are appropriate where a user is assigned exclusive use of a terminal for a fixed and regular period of time such as a shift. Magnetically encoded cards are appropriate where multiple users share terminals.

Authorization

Once users are identified and authenticated to an acceptable level of risk, they should be restricted to an appropriate set of resources. In general, since the rules are to be applied at the interface, they should be expressed in terms of the resource or service delivered at the interface. For example, at the interface between an operator and the tape library, the rule will be expressed in terms of tapes and may be something like "all tapes authorized by management" or "all tapes required for jobs scheduled".

An authorization will relate a user, a resource, and a rule. For example, USERA may have access to DATASETA for READONLY. Notice that these are "names" of users, resources, and rules, not the objects themselves. There will usually be at least one rule for each class of resource and often several. The rules will be appropriate to the resource. For resources like data sets, the rules will include such things as CREATE, READ, WRITE, UPDATE, APPEND, SCRATCH, and RE-NAME. For programs, the rules may include CREATE, MODIFY, and EXECUTE. For transactions, the rules may include AUTHORIZE or PERFORM. While the name of the rule is simple, the rule itself may be extremely sophisticated in application.

A user should be restricted to a set of resources that provides an acceptable level of risk.

The authorization function implies a data set associating users, resources, and rules of access.

Delegation

The delegation function is the mechanism for establishing and maintaining the rules of access and the authorization data set. In order to achieve an acceptable level of risk, this function must be highly responsive to the individuals who authorize users to resources (normally the user's immediate management). If it is not responsive, management will define broad (high-risk) rules in order to prevent the authorization mechanism from interfering with other benefits provided by the system.

The required sophistication of the delegation mechanism is a function of the complexity of the relationship between users and resources, the number of users and resources, the number of managers to be served, the number of locations, and the amount of activity. Thus, for simple relations between users and resources a small user population, a small number of managers and locations, it may be responsive to have a single staff person (security administrator) update the rules in batch mode. However, if there is a great deal of activity, it may be necessary for managers to be able to update the rules in a realtime mode.

Restrictions as to allowable associations may be enforced by the programs but at least one person must have the ability to override. For example, the general rule may be that those who are authorized to add vendors to the payable file should not also be allowed to approve invoices for payment. The delegation mechanism can be used as a tool to apply this rule. In a larger system, the delegation mechanism may be structured to disallow it. However, the system should not be so constructed as to make exceptions to the general rule impossible, because exceptions do occur and the system must be able to respond to them.

The delegation function is itself a resource and access to it must be restricted by the authorization mechanism. It may also have to exercise constraints over its users. For instance, it may be appropriate for the author of a data set to grant access to another person within his department but not to a user in a different department.

Journaling

Having identified the user, authenticated him to an acceptable level of risk, restricted him to a specified set of resources, and provided the ability to update the specification, the next step is to record transfers, transforms, and uses of resources.

In order to ensure that the system is adequate to fix accountability for all significant events to the level of a single individual, a record should be made of all uses of a resource. Journaling should take place at all interfaces to the system and at interfaces between subsystems where significant resource passes from one person's control to another's. In general, the use of a resource should be communicated to the owner of the resource and the manager of the user.

It is also useful to record security events such as changes to the authorization rules or variances from those rules such as attempts to access a secured file which failed because of improper authorization information.

Variance Detection

The system should include a variety of mechanisms designed to alert management to variances from the expected behavior, use, or content of the system. To be most effective, these mechanisms must communicate with the management that is best able to determine what corrective action is required and to initiate that action. These mechanisms may include action in realtime, such as locking terminals or sending messages, and journal postprocessors designed to improve management visibility into the journals. Variances from security rules should be communicated to the security staff, the owners of any resources involved (for example, terminals and data sets), and the manager of the user.

Summary

The six interface controls discussed above should be implemented in hardware or software at highly populated or active interfaces. They may be implemented by manual or administrative procedures at sparsely populated or relatively inactive interfaces. They should be provided by each subsystem in such a way that it controls its own domain. All of the functions across all interfaces should permit management to make appropriate trade-offs between security and other utilities such as usability, flexibility, performance, and cost.

These concepts are intended to be useful in design of systems that can be operated with an acceptable level of risk. Once the system has been implemented, it is critical that it be so operated. The remaining chapters will deal with this critical task.

Chapter 4. Traditional Classes of Control

Controls and procedures have traditionally been viewed as including:

Organizational Controls
Personnel Procedures
Operational Controls
Development Controls

They all can be effective in reducing risks.

Organizational Controls

The objectives of an organization structure will normally include effectiveness, flexibility, responsiveness, economy, and efficiency as well as control or security. Security complements the other objectives more than it conflicts with them.

The standards for organizational control are similar for manual or automated systems. However, the concentration of resource in the data processing function requires special consideration.

Separate DP from Users

Because of this concentration, DP should not report to any of its customer functions but to a common level of management. For example, it should not report to Payroll or Accounts Payable and under most circumstances it should not report to Finance, Engineering, or Manufacturing. By the same reasoning, no business transaction should originate within the DP function. The DP manager should be able to demonstrate that all work that he does and all resources that he consumes are authorized by, and on behalf of, an independent customer.

Transactions should be originated and source documents created by using departments. Using departments should also establish controls enabling them to determine that all work that they submit is processed and none is added (see operational controls below). This rule is particularly true for development of new applications or changes to existing ones. Application development is the single most sensitive application on the system because it is the means of controlling all others. Therefore, all requests for development of new programs or changes to old ones should be authorized, tested, and accepted by using departments.

Separate Duties within DP

In addition to separating the DP function from its users, separation of duties within DP must be maintained. To the degree permitted by scale, each of the following functions should be performed by a separate person:

Data entry (e.g., keypunch)
Operation – job initiation
Operation – data input (e.g., tape mounting)
Operation – data output (e.g., printer operation)
System programming
System library maintenance
Application design
Application programming
Program testing
Data definition
Library management
Scheduling
Output distribution
Maintenance programming
Management

These functions should not all be performed by a single individual. Even if only one person is associated with a system, the management function for the system should be exercised by a separate individual.

Special note should be taken of the fact that a number of the functions shown as needing separation are within the traditional role of the programmer. Traditionally, programmers have initiated and made a change often without explicit management authorization. Usually, one programmer has been responsible for writing, testing, and maintaining a program. Often, a single programmer wrote both the data definition section and the procedure section of a program. Frequently, the same programmer wrote both the file maintenance and transaction processing portions of an application. Not infrequently, a programmer could cause his own program to be executed against data that belonged not to him but to the organization. This kind of organization is not wrong in all cases, but should always be suspect since it may make it easier for programmers to convert an application to their own use. Management should consciously determine that the risk is commensurate with the efficiencies.

Maintain Traditional Separations

Finally, traditional controls and separations must be maintained in user areas. These separations will include separating transaction origination from approval and recordkeeping from custodianship. Origination and application of file changes should be separated from origination of transactions and application of transactions against the file. For example, the individual who adds vendors to the payables file should not be the same one who approves invoices for payment.

Assign Individual Security Responsibility

In general, security responsibility should be assigned in such a way that employees know what resources they are to protect and from what hazards; what variances they are to note and what corrective action is to be taken. For example, operators should know that they are to protect all data in their custody from modification, destruction, or disclosure, that they are responsible for noting data or machine anomalies or unusual occurrences in the machine room (e.g., a stranger) and that they are to call these variances to the attention of management.

Use Appropriate Span of Management Control

Risk is a function of the span of management control (number of people reporting to a single manager). In general, the more specific and similar the task to be performed the wider the span of control can be. Thus the span of control in data entry, where jobs are routine and similar, is typically wider than that in computer operations, which is wider than that in application development, which in turn is wider than systems programming.

In general, the span that is required to be able to detect and correct errors on a timely basis is usually narrower than that required for any other purpose. Since errors are the greatest threat to data integrity, what is appropriate for errors is adequate for other threats.

Another useful test is the amount of attention required to supervise people. The span of control in this case should be narrow enough to provide assurance that employees are doing what they are intended to do. This span is generally adequate to ensure that they are not doing anything else.

Use Security Staff Where Appropriate

When a manager is responsible for many or valuable resources, it may be useful for him or her to have a security staff. Normally, this staff will assist in planning and organizing to meet security objectives. It may assist managers in performing risk assessments and preparing emergency, backup and recovery plans, or recommend guidelines and standards for such plans. It may suggest the allocation of resource necessary to security objectives. It should note all variances from good security practice and recommend corrective action. But it is important to recognize that line management, not the security staff, is responsible for security. Staff is no more responsible for security than it is for profit. Security responsibility must be in the same hands as the resource to be protected and the authority to implement protective measures and take corrective action.

Personnel Procedures

Personnel controls are both important and powerful. Their primary security rationale should be to protect the employee from unnecessary failure, temptation or suspicion. If properly designed, they will also have the effect of deterring, limiting, and detecting malicious acts.

Hiring Procedures

A control objective in hiring procedures should be to hire individuals who are sufficiently competent for the task to be performed. This can result in the lowest cost over time by reducing errors and training cost. It will also help to avoid frustration because of inability to perform assigned tasks.

It follows that hiring procedures should provide as much information on qualifications (such as education and/or experience) as is necessary to make a judgment on competence for the task to be performed. Since this data may be costly to acquire as well as sensitive to disclosure, only data that is relevant to the decision should be collected.

In certain industries or institutions it may be required by law to acquire information relative to previous criminal convictions. Because of its sensitivity, this kind of data should be acquired only when there is a demonstrated need. It should also be used with great care since it is often incomplete and, even if accurate and complete, it may be a poor predictor of future behavior. The lack of negative data may demonstrate that the prospective employee has never done anything wrong, or only has never been caught. The presence of negative data may demonstrate that a person is inherently dishonest, or only was once in a position of such need and temptation that he or she committed a dishonest act and was caught.

The decision may be that the risk of hiring is too great to be accepted, or that the individual be hired only for a job that is closely supervised.

Control Vacations and Job Rotation

Vacations serve to reduce errors by improving morale and reducing fatigue, and job rotation reduces risk by improving the level of cross training. Also, it is conceivable that a policy of mandatory vacations and job rotation will deter some fraud, as the probability of discovery increases when the perpetrator's job is taken over by someone else. It appears that more embezzlements are detected by this control than by any other (frequently discovery of embezzlements detected in this manner is credited to "accident"). Many more are probably deterred by the potential embezzler's perception that he will automatically be detected at vacation time.

Rotation of assignment is particularly important for management and supervisory personnel. Experience indicates that, over a period of time, managers often become careless in their application of controls. A large number of embezzlements are perpetrated by very long-tenure supervisors, who had resisted reassignment or even promotion. On the other hand, there was usually high turnover among the individuals reporting to them.

Restrict Employee Accounts

While employee accounts do not represent a major area of loss, they are significant because they represent a major and unnecessary temptation for DP personnel.

It seems that data processing personnel are most likely to steal computer time since that appears to them to be the most convertible asset to which they have access. In fact, there is probably highly marketable data to which they have access, but they often fail to recognize its value. They do not have access to other readily convertible resources. But it is also possible for DP employees to convert other resources to their own use simply by crediting their account. Restricting employee accounts will reduce this exposure.

The general rule should be that personnel should not have an account in a ledger over which they have any control. However, there are normal exceptions to the rule. An employee must be assumed to have an account in a retail ledger even though measures are taken to reduce this probability. An employee may have to have entries in compensation and benefit ledgers. Each of these exceptions should be compensated for by additional management supervision.

Follow Complete Termination Procedures

If an individual's employment is terminated, among the steps that must be considered are the following:

1. Collect all identification including badges, ID, and business cards (new business cards and ID cards indicating retired status may be considered for retiring employees).
2. Revoke all powers of attorney including bank signature cards. Change or revoke all codes or passwords to which the employee was privy (note that the requirement to be able to do this must be considered when selecting the strategy for assigning passwords).
3. Collect all keys (including magnetic stripe cards), signature plates, and other evidences of authority.
4. Settle all accounts including expense accounts and courtesy accounts.
5. Reconcile accounts of any resource over which the employee had control, such as petty cash, parts inventory, or tape library. Where indicated for the protection of the employee who will assume accountability, an audit should be considered.
6. Reclaim all proprietary information in the custody of the employee.
7. Remind the employee of any ongoing contractual obligations to you, including restrictions on use of data to which the employee has became privy in the course of employment with you.

Operational Controls

Operational controls provide for the accuracy and integrity of data. They assist in the detection and correction of errors and help to ensure that all data is processed. Incidentally, they help to limit fraud since a system which provides for the detection and correction of errors can be expected to assist in detecting frauds.

Compare Output with Input

Most techniques for detecting errors are methods of comparing output with the input that generated it. The most common example is proofreading, which is used where the context is likely to indicate whether a character is correct. Another example is key verification, in which source data is key-entered twice. The entries are mechanically compared keystroke by keystroke, and variances are flagged for later reconciliation. This method is expensive, but works well even when the context gives no clue as to the correctness of a particular character.

Use Conversational Feedback

In online systems, conversational feedback is often used. The user strikes keys stimulating the system and the system responds in a manner designed to help the user detect and correct errors.

This takes place at several levels. At the elementary level, the user strikes a key and the system responds by printing or displaying the corresponding character. If the character is not the one the user expected, an error has been made and corrective action is indicated.

At a higher level, the system may compare data received with its expectation and give an alarm if it detects a variance. For example, the system may say "ENTER CUSTOMER NUMBER" expecting a seven-character numeric string in response. If it receives a six-character string or an alphabetic string, it may respond with an error message such as "UNABLE TO PROCESS RESPONSE. INPUT FORMAT NOT CORRECT." If the input format is what the system expects, it may test further to ensure that it has a master record associated with the number it received. If it has no record, it might respond "UNABLE TO PROCESS RESPONSE. NO RECORD FOUND".

If the input meets all of the system's expectations, it will still feed back associated data to the user. In response to a customer number, the system might feed back customer name and address, asking the user to confirm that this is the name and address that he expected. At the end of an extended dialogue, the system might feed back a summary and ask the user to confirm that it matches his expectation. Even the summaries may take place at several levels. There may be one for a transaction and another for a group or "batch" of similar transactions. These summaries are analogous to control totals in batch systems.

Experience with conversational feedback suggests that frequently errors will be detected. Since the detection is timely, the error usually is inexpensive to correct. The requirement that the user give a positive indication of the final correctness of the transaction helps deter him from deliberately entering data that he knows to be wrong.

It is useful to note that in conversational feedback systems the users actually check on the programmers and thus provide a very powerful control mechanism. For the mechanism to have maximum value, users should be encouraged to call to the attention of their management any unexpected system responses.

Use Redundant Data

Anytime data is moved or transcribed there is some risk that it will be lost or garbled or that personnel who control the movement will add or change data for their own purposes. In order to reduce these exposures, redundant data may be added to aid in the detection of loss of data or undesirable data. Parity checking is a primitive example. One bit is added to each group of six, seven, or eight bits so as to make each group adhere to a rule such as "odd" or "even" count. Any single-bit error will result in a variance from the rule. More sophisticated techniques such as "check-digits" or "hashing" codes enable the detection of multiple-bit errors and even the automatic correction of some errors.

Use Control Totals

Control totals are another example of redundant data. They are sometimes used at the message level, but more often at the application or transaction level. The control total is recalculated at the end of each transmission or processing step. A variance from one step to another may be evidence of an error. The control may be transmitted with the data, but comparison of its transmission at both ends will reduce the risk of someone modifying both the basic data and the control.

Control totals can also be viewed as a gross comparison between output and input. A total is taken of one or more fields across all transactions, and recomputed during each subsequent process. Agreement with the original number is evidence that no transactions have been lost or added.

For financial transactions, totals should be taken on at least two fields, account number and amount. This technique provides evidence that no transactions have been lost, added or misdirected (account numbers changed).

Taking control totals at intervals can assist in localizing any errors. Control totals taken across files or ledgers can be used to detect additions or deletions of records or accounts.

Control Error Corrections

The ability to detect errors implies the ability to correct them. Often this error-correcting function has more flexibility and less control than the function for recording transactions in the first place. Because of this additional sensitivity, access to these error-correcting capabilities should be restricted and additional management supervision should be automatic whenever they are used.

Supervise Recovery and Restart

The ability to recover and restart a job protects against the loss of resources invested in the job prior to a failure. Because jobs can fail in a wide variety of unpredictable ways, recovery and restart capabilities must be very flexible. As has already been suggested, flexibility is synonymous with sensitivity. Therefore, recovery and restart capabilities should be invoked only under management control and the sensitivity should be compensated for by additional management supervision.

Control Forms

Forms are often used as security mechanisms. Stock certificates, currency, preprinted checks, and savings passbooks are examples. The form itself may be the primary evidence that establishes the validity of information or the authority of a transaction. For this reason, most forms, including input forms, should be sequentially numbered and individually controlled. Sensitive data printed on otherwise blank stock should be page-numbered. When possible, a positive indication of the total number of pages should be printed with each page number, or at least an indication should be printed on the last page. For multiple-part forms, each copy should bear its own preprinted copy number as well as the total number of copies (e.g., copy 3 of 4 parts). Then, if the operator, for his own purposes, runs a job calling for 4-part forms on 5-part forms, this would be apparent to the installation's output control function and to those to whom the report is distributed. Consideration should be given to putting all forms, perhaps even including blank stock, under control of someone other than the operator.

Control Job Input/Output Media

In order to be in a position to fix accountability, it is desirable that input media be passed hand-to-hand. The receiver should ensure that the name of the person transmitting input to him is associated with that input. For example, when receiving a job, an operator should ensure that the name of the submitter associated with the job is the same as the name of the person from whom he received it. The common practice of having users drop jobs into a basket is risky and should be discouraged.

Control should also be maintained over output media to ensure that it is delivered to the person to whom it is addressed. For most data, normal mail is adequate for this purpose. For sensitive data, signed receipts may be required. Some installations use locked mail boxes for this purpose.

Notice that if USERA submits a job in the name of USERB, USERA may be found out if controls exist ensuring that the output media is delivered to USERB. USERB will receive unexpected output, indicating that input may have been submitted by someone else. Users should be instructed to call unexpected output to the attention of management.

Development Controls

In general, the objectives for development controls are to ensure the development of effective systems and programs and their effective maintenance. The security objectives of those controls are to ensure that the development process and the product system operation have an acceptable risk and that resources provided for development are not converted to any other purpose.

It is helpful to view the development activity in phases. A typical software life cycle might be broken down as follows:

Conception
Planning
Design
Programming
Testing
Implementation
Acceptance
Operation
Review and maintenance

Each of these phases should have a clearly specified and identifiable product, which should conform to standards for that phase. The product of one phase should describe the product of successive phases as well as the resource that the successive phase can be expected to consume. For example, the product of the planning phase is a plan. That plan must clearly describe the product specification that is expected as the product of the design phase. It should also state how much resource the design phase will require. The product specification must clearly describe the program that is the product of the programming phase and the test cases used in the testing phase. Both the product of a phase and the resource that it consumed should be reconciled with standards, good practice, and the expectation documented by the previous phase.

The programming phase may not be the most sensitive phase, but it is the one that stimulates the most concern. It is therefore appropriate to devote special attention to it.

View Programming as an Application

Computer programs are developed by procedures that do not greatly differ from other computer applications. However, they are highly sensitive since they involve flexible access to the system and since they are the means by which other controls are communicated to the system for it to enforce.

Unlike other applications, programming did not exist prior to the installation of the computer, and consequently there were no applicable pre-existing controls or even traditions of control. As a result, this highly sensitive application was often undercontrolled. The associated risk was acceptable as long as the population of programmers was small and the resource delivered to them was only a small computer and the data of a single job. In today's multi-programmed data base systems, however, rigorous controls are appropriate.

The principal objective of these controls is to reduce the risk to product performance, budget, and schedule. The security objectives should be to reduce error and the probability that a programmer will convert either the system, the data, or the application to his own use.

View Program Library as a File

If programming is viewed as another application, then it follows that the program library is its file. Controls over changes to this file should be like those for any other file, i.e., the process by which the file is changed must conform to good practice and the content of the file must agree with expectations.

Involve Multiple People in Changes to Library

The ability to make changes or additions to the program and the ability to authorize those changes should be assigned to separate individuals. Access to the system should be restricted and duties assigned in such a way that multiple people must be involved in making changes to the file. Again, it is the management of using departments who should authorize all new development and changes. The development or programming manager should also be required to approve.

Control Access to the Program Library

All changes and additions to the production library should be made by a single individual. Normally this individual should not perform programmer or user functions. He should perform his job in such a manner as to be able to demonstrate that all changes were authorized.

Access to program libraries should be controlled by lockwords or similar mechanisms. Consideration should be given to using software that provides control over access to individual library members as well as to the library itself.

Log All Changes to Program Library

A journal should be kept of all changes to the program library. This should reference the authorization for the change in such a way as to fix accountability to the programmer and to the approving managers. Additions or deletions noted in the journal should be reconciled with the authorizations by count. Often, the same software that provides access control over library members may be used to make journal changes to those members.

Structure Library and Fix Responsibility

The program library itself should be structured in such a way as to improve management visibility and control. It should be divided into sublibraries, programs, subprograms, and modules. Naming conventions should reflect logical associations of programs. For example, payroll programs may be kept in a sublibrary reserved for them and their names may all be prefixed by the same character string. Responsibility for each program should be clearly assigned to only one person at a time.

Use Control Data on Program Library

Record (program) counts should be used so that the loss or addition of programs will come to management attention. Instruction counts on source libraries and byte counts on object libraries can be reconciled with previously determined counts to deter and detect unauthorized changes.

These controls over changes to the program library are similar to those for changes to any other file. The techniques for reconciling the library to the expectation for it are more individualized, but they are still analogous to other files.

Specify Content of Library to Facilitate Reconciliation

The expectations as to the content of the program library must be expressed and reconciled at several levels. There must be a count of the number of programs by sublibrary; the number of programs in the sublibrary yesterday plus additions less deletions should equal the number in the sublibrary today. The total in the library should equal the sum of the sublibraries. Utilities can be used to reconcile daily the actual number of programs with the expected number. A list of programs by sublibrary might be reconciled with the actual programs in the library monthly or more frequently if there are variances in the count. Variances between the expected and the actual lists should then be reconciled with the journal of changes. This level is analogous to the reconciliation of the count of vendors in the payables ledger.

Compare Additions to Library with Authorizations

On a sample basis, the actual content of some additions should be compared with the specification referenced in the authorization. This is analogous to comparing a new vendor in the payable file with the authorization and source data for that vendor, but it is more difficult because the specification is more complex than the source data for a new vendor and its relationship to the program is not as simple as that of the source data to the vendor record.

For example, in reconciling the vendor record, it is helpful to proceed field by field. This is facilitated by a format for recording the source data and a record layout. It is also necessary to be able to display the record or to have a printout of it. The program should be reconciled with the specification section by section, e.g., environment, data description, procedure. The more highly the program and the specification are structured and the more parallel those structures are, the easier and more effective the reconciliation will be. As with the vendor record, facilities must be available for displaying or listing the program from the library.

Compare Programs with Expectations

Occasionally it is appropriate to compare the vendor file or the source data with the external environment. This ensures that the source data is not only complete and authorized to be in the file, but also represents a genuine vendor with whom legitimate business is being done. Techniques for making these tests include reference to directories such as telephone, industry, city or credit directories; mailing of letters of confirmation or statements; and finally, calls or interviews.

Similarly, it is not always adequate to know that a program has been written and tested according to procedures, is authorized to be in the file, and appears to conform to standards. It may also be necessary to ensure that it does what is expected and does not do anything else.

The rule that a program must "do what is expected" is much narrower than the rule that the program should "do good things" and, therefore, more variances must be reconciled. On the other hand, the "do good things" rule is so broad as to make the detection of variances difficult if not impossible. Therefore, it is recommended that the "expected" rule be used to detect variances, which may then be reconciled with the "good things" rule.

The "expected" rule is appropriate for all systems. However, there is a set of limited application systems for which a short but exhaustive list of all "bad things" can be defined. Where it results in fewer variances to be reconciled, it may be appropriate to use the rule "doesn't do bad things".

The expectation as to what the program will do should be documented in the specification and in a set of test cases with associated results. It is usually also necessary to make additional tests to ensure that this documentation accurately and completely reflects the intention of management and the expectation of the user. It should be apparent that changes to this documentation must also be carefully controlled. Test data is an important part of the documentation of the program. Test data should normally be prepared before the program is written. It should contain examples of all classes of input including "all others" (for example, unanticipated cases).

When this documentation is adequate, it is a relatively simple task to test that the program agrees (when the documentation is not adequate for this purpose, then it is inadequate by definition, and management action should be taken). First, the program is run against the test data and the results are reconciled with the expected results. Then a test is made to ensure that the test data exercises all of the paths and instructions in the program and again variances are reconciled.

Experience suggests that variances between actual and expected test results are caused by inadequacies in the specification or the test data as often as by errors in the program. This may result from the fact that the program is stored on magnetic media with powerful support for updating while the specification and test data are on paper media. The updating, use, and control of the specification and test data will be facilitated by keeping them on the same media as the program.

Compare Program Changes with Authorizations

Changes to existing programs within the library are analogous to transactions against other records, e.g., an invoice approved for payment. Similar techniques should be used for controlling them. Since changes are usually smaller but more frequent than new or deleted programs, the techniques should be adjusted accordingly. For example, the length of a new program may make it impractical to examine it exhaustively even though the low frequency makes it practical to examine all new programs in part. On the other hand, a change should be examined exhaustively even though all changes are merely sampled.

Compare Library and Programs with Earlier Version

Because of the frequency of changes, it is often useful to compare the library or program with an earlier version and reconcile variances with the change authorization. This is a particularly useful technique for an auditor to use for the purpose of determining the most useful programs to examine.

This discussion has emphasized control of programming. Control for the programmer is dealt with in the next chapter.

Chapter 5. Functional Duties

This chapter suggests a model for determining the controls that are appropriate to any position. Several typical data processing positions are then analyzed in terms of the model.

Step one is to examine the job description in order to determine the tasks to be performed. Next, determine the resources that are required to perform those tasks. Then establish controls to hold the individual accountable for each resource. The controls should be adequate to establish to an acceptable level of risk that an individual did what was expected and did not do anything else. While the model is simple, it must be consistently applied if it is to be effective.

Computer Operator

Consider, for example, the job of computer operator. The operator's job is to run those computer jobs indicated by management, to note anomalies in the running thereof, and to take the indicated corrective action. To run jobs, there must be access to the input and the output data for those jobs as well as adequate computer time; there need not be access to data for other jobs. Access to sensitive forms may also be required, but access to blank input forms or to output forms for jobs other than those to be run are not needed. The computer operator must have a high degree of access to the computer (a common, but for purposes of this discussion inadequate, description of his job is that he runs the computer). In order to note anomalies in the run, there must be access to all kinds of indicators including the console and the printer. In order to take corrective action, the operator's access to the computer must be extremely flexible.

To say that someone did what was required is to say that the product agrees with the expectation. In the case of the computer operator, this means "the output is the proper function of the input". To ensure that the operator did what was intended, compare the output with the input.

In order to ensure that the operator did not do anything else, account for all of the resource. Reconcile both the input and the output returned with the input received, the forms used to the input, and the forms remaining to the forms available less the forms used. Normally, these comparisons are made as the work and resources pass to or from the operator and those with whom he interfaces such as schedulers, users, or tape librarians.

Finally, account for the most convertible resource in the hands of the operator – computer time. At a gross level, this is the reconciliation of time available with time used plus time remaining. For some purposes, this accounting may be adequate. However, since a computer does large quantities of work in small increments of time, it is usually necessary to be more precise. Records such as the console log, system journals (e.g., System Management Facility; DOS/VS POWER) and production work orders will be useful for this. The jobs run, as indicated by the system journal, should be reconciled with the anticipated jobs as indicated by the production work order.

It is often alleged that these records are inadequate for this purpose, since the flexibility of access provided to the operator could permit their defeat. However, in the process of defeating the journal mechanism, he will leave variances in the journal. Even in the best designed systems these variances may not be easy to detect or obvious in their intent, but they will be there.

Application Development Programmer

Application development programmers are responsible for defining data and programs in accordance with user specification and management direction and for preparing test data and testing the program. Normally, this involves access to a test system, test data and test output, but not to a live system or live data. It is their responsibility to see that the program performs according to specification (and does nothing else) and that the test system is used only for the purpose for which it is intended.

Integration and user testing should be the primary technique to ensure that the program does what it is intended to do. Where the specification is adequate, this task is not too difficult even for large systems.

As a further safeguard, the code should be compared with the specification by the development manager or his designee. This comparison need not be done for 100% of all programs, but most programs should be sampled. The principle purpose of this review should be to ensure that the program performs according to specification and that it adheres to standards, but there should be no code that is not required by the specification.

In the past, this comparison has been difficult for a number of reasons: lack of rigorous or formal specifications, poor documentation, and programs that were large and unstructured (or whose structure did not match that of the specification) making it difficult to discern their intent or compare with the specification.

Modern programming technology (for example, chief programmer teams, top-down development, structured programming, HIPO documentation, and structured walk-throughs (see *Improved Programming Technologies – An Overview* GC20-1850) was developed to help address just these shortcomings of the product produced by programmers. It provides the mechanism for formal, rigorous, and consistent specification and good documentation. It results in small program modules whose structure follows that of the specification. It produces readable code, with manifest intent, readily compared with its specification. This technology was developed primarily for reasons of efficiency and reliability, but its greatest contribution may be to help provide control.

Maintenance Programmer

Maintenance programmers are responsible for changing programs in accordance with management direction in order to respond to changes in the environment. They are also responsible for determining problems and for recommending corrective action. If their recommendation involves a change to a program, they usually make the change.

The resources required are the same as those for an application development programmer except that the maintenance programmer may also require access to live data (for problem determination), to a copy of the program (but never to the line program), and occasionally to the live system (for problem determination and test). Any dumps or listings that may contain live data are normally the property of the using department. The using department should make the determination as to what part, if any, of that data should be seen by the programmer. The more data the programmer sees, the easier it will be to identify the problem. In most cases, the access of the programmer to the data will present no problem, but a judgment should be made in all cases.

The programmer should test the change and the user should test the changed code as well as the old code (regression test) to ensure that it conforms with the specification as changed by the change order. This testing demonstrates that the programmer did what was intended. To be certain that the programmer did not do anything else, development management should reconcile the new program with the old program and the change order. The amount of new code is usually short and readily compared with the change order. Programs are available for comparing the unchanged code from module to module.

If there is extensive new code, the concept of "throw-away-code" should be considered. This states that where the amount of work involved in a change exceeds a certain threshhold (one company uses two work-weeks), the least-cost, least-risk strategy is to throw away the old code and the old specification and start over from scratch. It should be kept in mind that the maintenance programmer (even though the author) does not understand the base program as well as the author did when writing it. As a consequence, there is a high risk that even a simple change may introduce a new problem.

Systems Programmer

The functions of the systems programmer may include the selection, generation, installation, testing, and acceptance of all system (nonapplication) software. In some installations, a systems programmer may make installation modifications to system software in accordance with management direction. This function may also be responsible for problem determination, documentation, and maintenance.

While the ratio of systems programmers to applications and maintenance programmers is usually small, the systems programmer job is very wide in scope. Systems programmers have responsibility for any system software that management may require to control use of the system, application programs, and indeed, the functions of all those who interact with the system. Since they have the ability to change system programs, then software itself is less reliable for the control of systems programmers. Therefore, it is particularly important that these functions be controlled by the kind of techniques noted in this section.

In order to do their job, systems programmers will require access to all input program media and documentation delivered from vendors. They will require machine capacity to perform system generation. Since they test systems programs whose anomalies may manifest themselves in hardware indicators, they may have to have access to the system itself for purposes of testing or problem determination. They may require access to some printed output from the generations and tests. Making modifications to system programs may require the ability to change program libraries. However, they do not need to have uncontrolled access to any of these resources.

All machine capacity that they use should be authorized by systems programming management and should be scheduled through, and delivered by the operations group. A management determination is appropriate for each instance in which a systems programmer wishes to be present for a run. Additional management supervision may be indicated. It is almost never appropriate for a systems programmer to touch the console, printer, input/output devices, or media. While anomalies may be noted by the systems programmer, all corrective action should be taken by the operator, and the operator should be required to report all anomalies and corrective action to his own management.

Normally, the products of a systems programming run should be the property of operations. Where it is appropriate for copies to be provided to the systems programmer, they should be provided through systems programming management.

To protect the systems programmer from suspicion and system data from the unexpected behavior of a test run, it is often appropriate to conduct systems runs in a "sanitized" environment. Where it is indicated, this can be accomplished by providing a virtual machine, by placing sensitive data offline under operator control, by scheduling a nonsensitive environment or even isolating the systems run in its own nonsensitive hardware environment. Care should be exercised that the expense associated with these measures is appropriate to the risks involved.

In controlling installation modifications to vendor code, the same controls are appropriate for a systems programmer as for a maintenance programmer. Care should be taken to control access to the program library, to keep a record of all changes, and to see that the change approved is the one made.

Management should satisfy itself that the use of the systems programmer's time, the use of system time, the product of all systems runs, anomalies noted, and corrective action taken agree with their expectations.

Magnetic Storage Media Librarian

The function of the librarian is to store, index and retrieve media on demand, but in accordance with rules whose purpose is to exercise control over the customers of the library.

In order to accomplish this function, the librarian must have access to the library and its contents. Notice that since the data is on magnetic media this access does not enable the librarian to see the content, but might allow the destruction of that content. This access implies that the librarian is accountable for all media and for adherence to the rules governing the distribution of media.

The library itself is a key technique for controlling the librarian. If it does not restrict access to tapes only to the librarian, then no accountability can exist. Therefore, it should be a closed facility with barriers.

Second, the librarian should be required to keep a log of all movements of tapes into or out of the library. The log of tapes out should reference the authority for delivering the tape (e.g., standing rule for scratch tapes, production work order for data tape). Reconciliation of the log and the authority can be used to demonstrate that the librarian did what was expected. (This same reconciliation might also demonstrate that an operator did nothing else.)

To demonstrate that the librarian did not do anything else, it may be necessary for management to reconcile the actual contents of the library with the expected contents. This will involve a physical inventory of the media and a reconciliation of variances using the logs, as well as testing to ensure that the index is correct and the media readable.

Conclusion

It may appear that these jobs are so narrowly defined as to be unrealistic. However, one way to reduce risk is to separate duties; the more narrowly defined a job is, the easier it is to control. An organization may not enjoy the economy of scale that would permit this kind of job structure. However, analysis using the model will still be useful and should lead to controls that are appropriate for that particular organization.

Chapter 6. Plans and Programs

This chapter describes programs designed to help high-level general management determine that all their line management is using consistent strategies and adhering to minimum standards in approaching its security objectives. The programs described are similar in that every department should be involved in them. Top management can establish the broad objectives to be achieved and allocate the necessary resources, usually including the establishment of a staff associated with the program. The staff can recommend guidelines, note variances, and recommend corrective action. The action under these plans can be taken by line management.

Risk Assessment

The objective of this program is to determine that all management has adequate visibility into its exposures. Managers are asked to catalog the resources in their care and the hazards to which they may be exposed. Economic consequences and expected rate of occurrence are then assigned to these exposures. The product of these numbers yields exposures in dollars per unit of time. While normally only line management can perform this analysis, staff is useful for assisting, for making certain that the analysis is done in a consistent manner, and for determining that each exposure is counted only once. This analysis may be a necessary precondition for the rational selection of protective measures.

Emergency Plan

Emergency plans are plans for responding to serious threats in such a way as to limit damage to critical resources and protect the mission capability. The emergency plan is separate from the plan to recover the mission capability should it be lost, or to accomplish critical portions of the mission while the capability is down (backup). Like most of the plans covered in this chapter, it should exist in all departments and at each level of management be comprehensive.

It should address natural threats, such as fire, wind, rain, rising water, and earthquake; and man-made threats, such as riots or bomb threats. The first priority should be the protection of people. While a different plan could be in force for each threat, it is usually more effective to have a comprehensive plan that addresses all threats. Since the objective of the plan is to protect the mission capability, strategies should aim at protecting resources in the order of their importance to the mission (not the difficulty of recovering them). Effective strategies will include some combination of communication of an alarm, evacuation or shelter of people and protecting machines with covers, drains, pumps or fire-quenching systems. Responsibility for making decisions as to the indicated action should be clearly assigned in the plan. The plan or appropriate parts of it should be posted or otherwise communicated to all employees. Drills should be held on a regular (though nonscheduled) basis, as they are useful both for training and for testing the plan. Records should be kept of all drills to enable top management to ensure that the drills are being held and to let immediate management detect the need for corrective action or plan adjustment.

Backup Plan

The backup plan deals with how critical portions of the mission will be accomplished between the loss of mission capability and its recovery. As was suggested earlier, all departments, including user departments, must have their own plans. In drafting a backup plan, it is useful to list some assumptions about the nature and extent of the loss.

Identify Critical Jobs

A critical job is, among other things, one that must run at a particular time. It follows that what is critical will vary with the calendar. For example, if the system fails on the day that payroll is normally run, then payroll is critical. On a different day, billing may be important. These examples also illustrate another possible characteristic of a critical job – it is important to the cash flow of the business. In many cases, the definition of "critical" may apply to about 10% of all jobs run. It is generally the user who determines which jobs meet this definition; only rarely will the DP staff have adequate information to make this decision. The user is also usually the best person to determine what alternatives exist.

Identify Critical Configuration

For each critical job, the critical system configuration required to run it must be identified. Some jobs are designed to exploit all of the characteristics of their home system and are therefore configuration-dependent. They are usually less portable than those specifically designed to be configuration-independent and portable.

Identify Backup System

The backup strategy will also involve identifying a system with a suitable configuration and sufficient available capacity to run the identified critical jobs. Ideally, this alternative capacity will be within the same organization, and all be in one installation, but in practice this will not usually be the case. Rather than rely on a single installation for capacity that represents a significant portion of its total, it is better to look to multiple installations.

Assign Backup Missions

Where an organization runs multiple installations, backup missions should be assigned. Part of the mission of installation A may be to have the potential to provide a certain amount of capacity to installation B. In support of this mission, an installation may have capacity that it uses only for readily displaceable jobs.

Assign Individual Roles

Personnel assignments are another critical portion of the backup plan. Management should assign, in advance, a primary and secondary individual to each critical job, informing them precisely what their role is to be. Care should be taken to ensure that backup assignments do not conflict with recovery assignments. An individual should normally be assigned to one or the other, though in small installations an employee may have to be assigned to both.

Provide for Security in Backup Situation

Special security strategies may be required in a backup plan, to determine that normal security is maintained to the degree possible and that additional protective measures, usually management supervision, compensate for the additional risks that will be encountered in secondary sites.

Test Backup Plan

In order to determine that the backup plan is adequate and workable, it should be tested. Since it may be disruptive to normal operations to test all applications at once, it is often desirable to test them one at a time. The tests should be designed to demonstrate the compatibility of jobs, job control language, operating systems, and configurations. The frequency of tests should be a function of the number of variances noted and changes in the mission or environment. If, in conducting a test, a large number of variances are noted, then another test should be held soon. If few variances are noted, then another test may not be indicated unless there are new applications, changes in hardware or software, or changes in the plan itself.

Recovery Plan

The objective of this plan is to recover the primary mission capability. It complements the emergency and backup plans, and like them, must exist in all functions and at all levels. The mission capability of a DP installation can be viewed as being made up of a combination of people, space, equipment, communications, data, and supplies. The recovery plan should identify these specifically, as they apply to the installation, and outline the strategies for recovering each one. In general, these strategies should involve identifying one or more alternative sources.

Make Assignments and Plans for Communicating Them

People are the most difficult resource to replace. Their experience, training, and knowledge make them unique to the installation. However, rarely are all the people associated with an installation subject to a single disaster. Because of the uniqueness of the individual, an installation should base its recovery strategy on the probable survival of some of the staff. Natural geographic dispersion will normally provide at least a partial complement of personnel to maintain the installation. The recovery plan should include the matching of recovery tasks and people.

The recovery strategy should also provide for management communication with all personnel. The plan should provide for informing them in advance of their role and responsibility in the recovery and for notifying them in an emergency of the nature of the problem as well as when and where to report.

Identify Alternate Sites

The space used for the DP mission has many special characteristics and properties. It may have large undivided expanses, exceptional power and cooling, raised floor, security, and in the case of large-scale equipment, chilled water. It may have to be located near the mission supported, and it may also be located sufficiently near the people who work in it. It is so unlikely that a particular installation will ever suffer a disaster that no organization can afford to set space aside to be used only for that purpose. The recovery strategy should identify enough locations as both suitable and available so that an acceptable level of risk is achieved. Large organizations using a great deal of space in the community generally have a facilities management staff. The DP staff need only apprise the facilities management staff of their needs. Smaller organizations in large communities may consider retaining a commercial realty broker to keep track of appropriate and available space. Small organizations in small communities may need to keep a list of usable sites on the assumption that one or more can be made available in an emergency situation.

Identify Source of Replacement Equipment

Today, the population of computer hardware, even of any given type or model, is quite large. Job streams are often portable from one system to another. Components are portable and can be readily moved from one location to another. Vendors are particularly responsive to emergency situations and usually have a detailed description of the installed hardware. As a result, most data processing installations can expect to be able to find replacement hardware faster than they can find adequate space to put it in.

However, certain situations may require special strategies. If an installation is dependent upon a specific unit known to be in short supply, there should be adequate provision to either locate an identical one or substitute some other component. If a unit has an installation time known to be greater than the objective recovery time, it may be useful to plan to recover with a different unit, perhaps with higher cost or lower capacity, but with shorter installation time. Finally, if an upgrade or reconfiguration is already planned, it may be useful to accelerate the schedule at recovery time. The recovery effort can be made to displace some of the planned change effort.

Plan for Replacement Communication Facilities

Communication facilities normally have long lead times, which can generally be shortened only slightly even in an emergency. These lead times should be borne in mind when determining a recovery plan. When the required lead time of the desired facility is greater than the objective recovery time, a strategy must be prepared to substitute a slower or higher cost but available facility. For example, dial-up lines may be substituted for leased lines, voice facilities for data facilities, or mail for telephones.

Often the availability of communications facilities varies with geographic location. For example, a suburb may have a modern mail facility while a rural community may not; the phone company may have more available capacity in a new central office than in an older one. If communication capacity is important to an installation, this geographic availability of communications may influence the choice of recovery sites. In any case, the situation should be known and planned for.

Provide for Availability of Data, Supplies, and Recovery Plan

Data stored on magnetic media is relatively inexpensive to copy, transport, and store. Therefore, an installation can and should have a special copy of data to use in recovery. The frequency with which this copy should be prepared is a function of the frequency and quantity of the activity. In addition to the data that is used by the system, it is also essential to have copies of programs, tables, documentation, and run books.

Supplies in reasonable quantities have fairly short lead times. The plan should indicate the vendor contacts. Samples of the forms are often helpful in describing requirements to the vendor.

Copies of the recovery plan should be kept at home by multiple managers to ensure that it is available when needed.

Test Recovery Plan

The recovery plan may be tested one strategy or resource at a time so as not to disrupt normal operation. Since in real efforts the recovery and backup will compete with each other for management attention and personnel resource, it may be wise to run a periodic test that involves both at the same time.

Vital Records Plan

A vital records program supplements the data section of the recovery plan. Its objective is to provide special protection to data that is essential to protecting the equities of employees, customers, stockholders, or the interests of society and to provide those records needed to resume operations. In identifying data to be covered by this program, the keyword is "vital" and the decision is binary. In a manufacturing organization perhaps only 3-5% of the records would be vital, while in a financial institution, the percentage might be higher.

Natural or specially prepared copies of these records should be stored offsite. Protected storage onsite probably is not adequate for this program, since even if the records survive, they might not be available on a timely basis.

The program should provide that the records are safe in this offsite location and updated on a timely basis. Written procedures, definitely assigned responsibilities, and schedules for vital records are necessary for the successful operation of the program.

Management should conduct appropriate tests to determine that records are sent out according to the schedules, that they can be retrieved on a timely basis, and that they are usable when retrieved. If staff is used to conduct these tests, appropriate reports should be addressed to management.

Access Control Plan

Access control limits risk by limiting the number of personnel who have access to sensitive resources. There is minimum risk when the number of individuals with access to a location is the same as the number who work in that location. The objectives for a program should be to come as close to this rule as is consistent with other objectives. All access that is not consistent with the rule chosen should be treated as extraordinary and compensated for by sponsors, escorts, and management supervision.

Localize Control

The more access control is localized, the easier it is to achieve. For example, while it is rarely practical to limit access to all of a public building to people who work there, it is possible to restrict access to a sensitive room within the building, such as the chief executive's office.

Use Multiple Control Points

The effectiveness of access control can be improved by having a number of control points – for example, the perimeter of the building, the entrance to the data processing department, and the entrance to the computer room. A compromise at any single control point will then involve less risk, and there is lesser probability that all three will be compromised.

Use Appropriate Technology

For small populations of people, control can be achieved by management supervision. However, for larger populations, more sophisticated strategies are indicated. In any case, the smaller the number of entrances, the better the control. For buildings, the number should normally be in the low tens and for sensitive facilities such as the mailroom or the computer room, it should normally be one. For populations numbered in the high tens, badges should be used. The larger the number of badges, the more information, such as signature and name, should be added to personalize the badge to the user. For populations in the tens of hundreds, photographs should be used. The larger the number of control points, the more controls such as color coding are required to distinguish the authority associated with a particular person. For populations in the tens of hundreds and control points in the low tens, an access control system that is magnetic badge actuated, computer controlled, and table driven could be considered.

Note Variances and Follow Up

An access control program will be only slightly more effective than its variance detection and follow-up provisions. Employees should note and report to management any strangers in their immediate work area. Management should note and report to the security staff employees without badges in their area of responsibility. Management should report to the security staff visitors without escorts.

The security staff should note and follow up on use of emergency-only doors, doors held open, and use of doors after hours. Security staff should note and report to managers employees without badges, use of badges at control points where they are not authorized, and entry after hours. The security staff may also conduct tests to ensure that strangers are being challenged.

Data Classification Program

The objective of this program is to label all media with the name of the set of protective measures that is appropriate to the data that is recorded on the media. It reduces risk by increasing the probability that data receives the appropriate protective measures and reduces cost by avoiding the overprotection of data. The classification is a statement about the amount of money to be spent to protect data, i.e., the amount associated with the set of procedures with whose name the media is labeled.

For example, the military may label a document "SECRET". The label is a direction that the procedures appropriate for "SECRET" are to be followed for that document. This may *include* a particular kind of background investigation that must be conducted for all individuals who will handle that data, but does not imply that all persons who have had such investigation may handle the data. Nor does it suggest that the sensitivity of that document to modification, destruction, or disclosure is the same as that for every other document so labeled.

The sensitivity of data to modification, destruction and disclosure is a function of many factors: quantity, association, interpretation, age, density, number of copies and media. In general, the sensitivity of data increases with quantity along an S-shaped curve rising at a decreasing rate for large quantities. The sensitivity of data rises exponentially with the number of distinct associations so that "employee number" and "salary" is more sensitive than either alone and "name" plus the other two is even more sensitive.

Data also becomes more sensitive with interpretation such that raw data is less sensitive than organized data, which is less sensitive than the conclusions drawn, which in turn are less sensitive than the plans of action that may result from the conclusions. With a few exceptions, the sensitivity of data decreases with age so that last year's payroll register is significantly less sensitive than today's. The sensitivity of data to conversion may vary with the density of the media because of portability so that listings may be less sensitive than tape which may be less sensitive than microfilm. Channel-width of a communications link is a special case of density. For example, more data can be compromised via a high-speed link than via a low-speed one.

The sensitivity of data to destruction goes down as the number of copies increases, but the probability of its disclosure goes up. Data on paper is sensitive to disclosure but modification or destruction leaves evidence. On the other hand, data recorded on magnetic media requires sophisticated equipment to read but modifications or changes do not necessarily leave evidence.

When deciding how to classify data, the consequences to the business of the destruction, modification or disclosure of the data should be considered in addition to the general characteristics of data and media. These consequences may be related to the cost of replacing or correcting the data, the cost of recovering or replacing property associated with or controlled by the data, lost revenue caused by the inability to render a service associated with the data, lost competitive advantage or opportunity, or the cost of compensating someone who is damaged by the compromise of the data.

Establish Classification Rules

Management should set forth the rules by which the authors or proprietors of data will classify it and the manner in which they should label it. The actual classification of the data should be done by the authors or proprietors since they have the necessary knowledge. The rules should be appropriate to the media on which the data is recorded. Since documents are normally more sensitive to disclosure, they are associated with procedures such as "Internal Use Only", "Proprietary", "Confidential", or "Not to be Reproduced". The name of the classification is normally printed on them. Since magnetic media is normally more subject to modification it may be associated with procedures such as "Sensitive", "Authorized Personnel Only", "Management Authorization Required" or "Two Levels of Authorization Required".

Restrict Classification Authority

The rules should avoid overclassification since it increases cost and reduces effectiveness. This can be accomplished by restricting the authority to assign expensive classifications. Since the sensitivity of data normally decreases with age, the rules should mandate termination dates for the classification, e.g., "Confidential until (date)".

Staff should note and report to management failure to classify, failure to assign termination dates, and overclassification. Normally, this will be done on a sampling basis.

Data Handling Policies

In support of the data classification program, there may be a set of broad data handling policies dealing with document security, waste disposal, distribution and use of reproducing facilities.

Keep Sensitive Data under Lock and Key

One such policy might provide that certain classifications of data must be attended or locked up, and may not be left unattended. Security personnel would note variances from these procedures and report to management. Reports might compare areas or activities. Corrective action might include collecting the document and making the holder call for it (from high-level management if indicated).

Provide for Secure Waste Disposal

Special facilities and handling procedures might be indicated for the disposal of sensitive media. These may include locked waste receptacles or shredders as well as the incineration or mulching of all waste documents. They may include appropriate erasing techniques for reusable magnetic media. Sensitive waste may be handled by teams or pairs of people often including security personnel.

Provide for Secure Distribution and Safekeeping

Special distribution or library facilities may be indicated for the most sensitive class of documents. Staff would distribute data in accordance with need-to-know lists, keep records of distributions, keep records for numbered copies and act as the control point for reproduction. They note and call management attention to unaccounted for copies or late return of copies.

Control Copying and Reproduction

It may be appropriate to control access to copying and reproduction equipment. All copiers should be labeled under the object stage so that the identification of the copier appears on all copy pages. In addition to its classification, some sensitive documents should carry a copy number. Where appropriate, a document should have an overprint specifying "DO NOT COPY". Finally, copying equipment may be attended. The responsibilities of the attendant could include enforcing rules, noting variances and keeping records (note that these procedures are not designed to stop the intentional spy who will bring his own camera, but the more probable proliferation of copies that results in accidental compromise of the data and loss of accountability).

Education

The business objective of the education program is to provide people with the skills and knowledge necessary to the performance of their jobs. It reduces the risk associated with failure to perform. It can make a more direct contribution to security by demonstrating to the employee the value of the resources within his control and the importance of his role in protecting them.

For this purpose, management will wish to consider establishing a formal education or communication program for security. This program may be implemented using some combination of the education, communication, and security staffs. However, the most important way management demonstrates its concern for and commitment to security is by the way it responds to a variance. If managers follow up on variances in a concerned and consistent manner, the employees will learn that they mean business. On the other hand, if managers tend to disregard variances, employees will follow their example.

Auditing Program

The auditing program is designed to provide management with a continuing and independent review of compliance with controls and procedures and into residual levels of exposure.

A full discussion of the role of the audit program in the control of risk is beyond the scope of this material. However, at a minimum, the auditor should be aware of all objectives and procedures for all risk reduction programs. They should note variances from them and recommend corrective action.

Chapter 7. Conclusion

This document has looked at the contribution of controls and procedures to security from several different perspectives. Its recommendations are primarily designed to ensure that people do what is intended and do not unintentionally cause harm by doing the wrong thing. This emphasis stems from the observation that far more damage results from the accidental acts or omissions of authorized and well-intentioned people than from all other sources.

Incidentally these recommendations will function to deter and detect intentional malicious acts on the part of authorized as well as unauthorized people, since a system that can detect and correct errors will also detect and correct intentional but false activity.

The procedures recommended are intended for normal operation. Often, procedures are criticized on the basis that they will be awkward in unusual or emergency operation. This sounds reasonable, but frequently results in weak, high-risk controls. Controls should be selected for their appropriateness to normal operation. Managers should have the authority to substitute emergency controls in emergency situations. This strategy will provide high security while maintaining the flexibility required to cope with an emergency.

While employee reassignments and discharges, suits under trade secret and patent laws, and prosecution under criminal codes are important corrective actions in a security program, the most frequently cited corrective action has been "inform management". This recommendation recognizes the fact that most people want to do what is expected. Variances are often associated with an imperfect communication of the expectation. For these reasons, management recognition is often sufficient and indeed the only corrective action indicated. Corrective action that goes beyond management recognition is so much a function of management style that more specific recommendations would be neither useful nor appropriate.

In order for security measures to be cost effective they must be applied selectively. Sensitive resources must get appropriate protection and, conversely, expensive measures must be applied only to sensitive resources. This concept should result in "rings" of protection and "islands" of security.

No system that supplies any other value also provides zero risk. Security is achieved only at the cost of some other benefit. The tradeoff of these utilities is perpetual. By the same token, the security program is never complete. It must be continually evaluated and adjusted to compensate both for its own shortcoming as well as changes in the environment.

Bibliography

The Considerations of Data Security in a Computer Environment (G520-2169), IBM.

The Considerations of Physical Security in a Computer Environment (G520-2700), IBM.

Guidelines for Automatic Data Processing Physical Security and Risk Management, U.S. Department of Commerce, National Bureau of Standards, Washington, D.C., Federal Information Processing Standards Publication (FIPS PUB) 31, September 1974.

Guidelines Establishing Requirements for Security and Confidentiality of Information Systems, Intergovernmental Board on Electronic Data Processing, Sacramento, California, 1974.

IBM Data Security Forum, Denver, Colorado, September 1974 (G520-2965), IBM.

IBM Data Security Symposium, April 1973 (G520-2838), IBM.

Improved Programming Technologies – An Overview (GC20-1850), IBM.

Management Controls for Data Processing (GF20-0006), IBM.

Martin, James, *Security, Accuracy, and Privacy In Computer Systems*, Englewood Cliffs, N.J.: Prentice-Hall, 1973.

Organizing the Data Processing Activity (GC20-1622), IBM.

Protection of Electronic Computer – Data Processing Equipment, Boston, Massachusetts, National Fire Protection Association, NFPA Standard No. 75, 1972.

Reed, Susan K. and Branstad, Dennis K., Editors, *Controlled Accessibility Workshop Report*, U.S. Department of Commerce, National Bureau of Standards, Technical Note 827, May 1974.

Reed, Susan K. and Gray, Martha M., *Controlled Accessibility Bibliography*, U.S. Department of Commerce, National Bureau of Standards, Technical Note 780, June 1973.

Study Group on Computer Control and Audit Guidelines, *Computer Audit Guidelines*, Toronto, Canada: Canadian Institute of Chartered Accountants, 1975.

Computer Control Guidelines, Toronto, Canada: Canadian Institute of Chartered Accountants, 1970.

Data Security Controls and Procedures — A Philosophy for DP Installations

READER'S COMMENT FORM

G320-5649-1

This form may be used to communicate your views about this publication. They will be sent to the author's department for whatever review and action, if any, is deemed appropriate. Comments may be written in your own language; use of English is not required.

IBM shall have the nonexclusive right, in its discretion, to use and distribute all submitted information, in any form, for any and all purposes, without obligation of any kind to the submitter. Your interest is appreciated.

Note: *Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.*

Possible topics for comment are:

Clarity Accuracy Completeness Organization Coding Retrieval Legibility

If you wish a reply, give your name and mailing address:

____

____

____

What is your occupation? ______________________________

Number of latest Newsletter associated with this publication: ______________

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.)

Note: Staples can cause problems with automated mail sorting equipment.
Please use pressure sensitive or other gummed tape to seal this form.

G320-5649-1

Reader's Comment Form

Fold and tape　　Please Do Not Staple　　Fold and tape

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS　PERMIT NO. 40　ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE:

International Business Machines Corporation
Department 824
1133 Westchester Avenue
White Plains, New York 10604

Fold and tape　　Please Do Not Staple　　Fold and tape

IBM®

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, N.Y. 10604

IBM World Trade Americas/Far East Corporation
Town of Mount Pleasant, Route 9, North Tarrytown, N.Y., U.S.A. 10591

IBM World Trade Europe/Middle East/Africa Corporation
360 Hamilton Avenue, White Plains, N.Y., U.S.A. 10601

Cut or Fold Along Line

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue
White Plains, New York 10604

IBM World Trade Americas/Far East Corporation
Town of Mount Pleasant, Route 9
North Tarrytown, New York 10591

IBM World Trade Europe/Middle East/Africa Corporation
360 Hamilton Avenue
White Plains, New York 10601

IBM®

Data Security Controls and Procedures - A Philosophy for DP Installations Printed in U.S.A. G320-5649-1

Printed in U.S.A. G320-5649-1